



A GAN based PID controller for highly adaptive control of a pneumatic-artificial-muscle driven antagonistic joint

Zhongchao Zhou¹ · Yuxi Lu¹ · Shota Kokubu¹ · Pablo Enrique Tortós¹ · Wenwei Yu^{1,2}

Received: 15 November 2023 / Accepted: 17 April 2024 / Published online: 4 June 2024
© The Author(s) 2024

Abstract

Upper limb prostheses are commonly propelled by pneumatic artificial muscles organized in an antagonistic arrangement. Nonetheless, the control of upper limb prostheses under changing/unknown situations is difficult and necessary for a variety of real-world applications. Adaptive control, learning-based control, and robust control have been studied to deal with such challenges. However, their adaptability is insufficient for prostheses used in daily life, which are exposed to variable task levels, user motor characteristics, and prosthetic features. This paper introduces a highly adaptive controller for the first time based on Generative Adversarial Nets and proportional–integral–derivative controller (G-PID controller). G-PID controller comprises a generator for generating compensation actions to enhance PID responsiveness when controlling the unknown/changing system. Moreover, it incorporates a discriminator that receives responses from both a user-preselected reference system and the compensated changing/unknown system, and simultaneously determines the source of these responses. Through continuous updates, the compensator modifies the response of unknown/changing system to align with the reference system, thereby facilitating adaptive control. The G-PID controller's effectiveness is evaluated through 1-degree of freedom (DoF) joint and 2-DoF shoulder prostheses in simulation experiments, and further validated in prototype experiments focusing on online learning for unknown and time-varying payload. The results demonstrate its ability to deal with diverse types of unknowns/changes, marking a significant advancement towards incorporating prostheses seamlessly into daily life.

Keywords Generative Adversarial Nets based PID controller (G-PID) · Learning-based adaptive control · Pneumatic artificial muscles (PAMs) · Antagonistic structures

Introduction

The loss of one hand can have a substantial impact on one's level of autonomy, restricting one's ability to undertake

employment, social, and activities of daily life (ADL) [1]. Upper limb prostheses enhance the quality of life (QOL) of amputees by physically and functionally replacing the missing or lost limb [2]. The upper limb prostheses are frequently driven by pneumatic artificial muscles (PAMs) configured antagonistically, as PAMs have superior mechanical performance, particularly a higher power-to-weight/volume ratio and a lower cost, which is critical for human–robot interaction [3–5].

Normally, the capability of the controller for robotic arms has an effect on their total performance in robotic control, particularly for the upper limb prostheses [6]. Two main challenges arise in the control of PAMs-driven upper limb prostheses:

- Firstly, highly nonlinear characteristics, significant hysteresis, low damping, parametric uncertainties, and time-variant properties of PAMs (largely attributed to the

✉ Wenwei Yu
yuwill@faculty.chiba-u.jp
Zhongchao Zhou
zhouzhongchao@outlook.com
Yuxi Lu
yuxi.lu@chiba-u.jp
Shota Kokubu
s.kokubu@chiba-u.jp
Pablo Enrique Tortós
pablotortos19@gmail.com

¹ Graduate School of Science and Engineering, Chiba University, Chiba, Japan

² Center for Frontier Medical Engineering, Chiba University, Chiba, Japan

compressibility of air) [7, 8], make them inherently challenging to control.

- Secondly, the control accuracy for each joint in PAMs-driven upper limb prostheses is easily affected by the mass of the payload [9] or external disturbance [10], both of which are generally unknown to the system, making it difficult to achieve real-time, high-accuracy control.

To overcome these challenges, some researchers have proposed the use of nonlinear robust controllers, such as sliding mode control [11], backstepping control [12], feedback linearization [13], and linear quadratic regulator [14]. While these controllers have shown the efficacy in ensuring the stability of systems and achieving tracking performance even in the presence of unknown parameters. In scenarios characterized by complex parametric uncertainties and external disturbances, static nonlinear robust controllers may not be effective in controlling structures driven by PAMs [15]. Therefore, the adaptive controllers with enhanced robustness capabilities are more suitable for simultaneously addressing these two issues. The adaptive controllers can be categorized into three types: model-based adaptive controller, online tuning adaptive controller, and model-free adaptive controller.

The principle of a model-based adaptive controller is to model the unknown parameters and conducting estimations or assumptions about these parameters. In [16], adaptive models have been employed for controllers, with least-squares algorithms used to identify system uncertainties online. In [17], the authors have designed an adaptive update law to compensate unknown parameters of PAMs through online estimation, and stability analysis is conducted using a Lyapunov function. In general, the model-based adaptive controller often employs Lyapunov functions for proving stability, thereby exhibits enhanced robustness [18]. However, it necessitates a highly accurate dynamical model, which is not always easy to obtain. Even if a model can be established, it is unlikely to account for all unknown variables or ensure that all parameter calibrations are correct during the modeling process and the phase of making parameter assumptions. This is particularly true for PAMs-driven system, under conditions of nonlinearity, time variation, and fluctuating air pressure [17]. Additionally, some methods involve modeling and identifying unknown parameters using bio-signals such as electromyography [19, 20]. However, calibration is always necessary in this procedure, adding to the method's complexity. Furthermore, the complexity of the such bio-signals sensors' wiring has an effect on the ease with which this technique can be worn [21].

Another approach to achieving adaptability is by online tuning of the controller's gains; the theory behind this method is that modifying the controller's gains can alter the system's response [22]. Anh has proposed a neural network self-tuning PID control (NN-PID) algorithm for the 2-axes

PAMs driven robot arm [23]. In [24], authors have proposed a fuzzy self-tuning PID controller of PAM to improve the control performance. However, online tuning adaptive controller is complex and the algorithm's implementation requires a significant amount of hardware resources [25]. Additionally, the ultimate control performance is still dependent on the controller, and there's no assurance that the tuned gains can consistently enhance the control performance.

The model-free (or data-driven) method is also used in the PAMs-driven structure to achieve adaptability. A data-driven multi-input multi-output discrete-time predictive model-free adaptive integral terminal sliding mode controller have been proposed to deal with system uncertainties for a PAMs driven lower limb exoskeleton system [26]. In [27], a controller have been developed for achieving highly sensitive position tracking with PAMs using a data-driven predictive control algorithm, adaptable to various scenarios. Overall, model independence and high adaptability are characteristic features of model-free adaptive controller [28]. However, model-free adaptive controller can be computationally costly and difficult to exploit in general [29]. Moreover, stability and robustness cannot be guaranteed for model-free adaptive controllers, particularly in the event of a sudden disturbance [30]. In addition, some model-free methods compensate the PID output using a neural network, which can be termed as feedback error learning control [31, 32]. Such method demonstrates specific characteristics: in the initial training stages, the PID controller serves as the primary controller to guarantee stability; in the later training stages, the neural network takes over as the main controller to ensure better control performance. However, the design of its loss function indicates the aim of substituting the PID controller with the neural network, which makes this approach unadaptable. In summary, traditional model-based, online tuning, and data-driven adaptive controller each have their shortcomings—those with enhanced adaptability often lack in stability and robustness, whereas those with improved stability and robustness may exhibit insufficient adaptability. Additionally, none of these controllers consistently ensure a satisfactory control performance. Thus, there is a need for an adaptive controller that can ensure stability, achieve high adaptability, and maintain better control performance.

Therefore, in this study, we present a controller based on Generative Adversarial Nets (GAN) and PID controller (G-PID controller). GAN has been introduced in 2016 by Goodfellow [33]. The principle of GAN is that given real data, and the generator creates synthetic data, while a discriminator evaluates whether the data is real or synthetic, thereby refining both the generator and discriminator through a min-max game. This iterative process aims for the generator to produce data indistinguishable from real data. The data generation capability of GAN offers the potential to maintain consistency between unknown system and known system.

Based on this principle, the proposed G-PID controller consists of the generator, discriminator, a reference response from a reference system, and the PID controller. Similar to the GAN process, the generator is utilized to generate the PID controller's compensation action. The compensation action in conjunction with the PID action as the unknown system's input, and the unknown system's response is set as a fake response. Simultaneously, the response of a reference system is selected as the reference response. The discriminator is presented with both the fake and reference responses and detects if the response originates from the unknown or reference system. The generator and discriminator are updated in accordance with the GAN update rule, and it is expected that the fake response of the unknown system can be compensated to align with the reference response, thereby achieving adaptability. According to our understanding, there hasn't been any application of GAN in the control domain aimed at achieving high adaptability.

The G-PID controller has the following advantages, especially for PAMs-driven system:

1. Because the G-PID controller is based on the system's state, it can be used no matter there is a model or not. Theoretically, adaptive control can be performed in any unknown (internal or external) condition. This characteristic effectively addresses the shortcomings of custom-made, wear-prone PAMs-driven systems, which often present challenges in terms of parameter identification.
2. The unknown system's fake response tends toward the given reference response, which means that not only can adaptive control be achieved, but also that the control performance can be predicted, ensuring that the response after compensation is predictable. Such an attribute significantly enhances the safety of PAMs-driven systems, which are frequently employed in human–robot interaction scenarios.
3. Due to the fact that the PID controller serves as the main control unit, its stability can be guaranteed throughout the control process, even in the event of a sudden disturbance, which makes the control system robust.
4. Although the value of compensation action required is unclear, as an unsupervised learning approach, the generator does not require any label throughout the learning process, which is not possible in conventional supervised machine learning.

Methods

Dynamic model for PAMs and PAMs-spring driven antagonistic structure

According to [34], PAMs can be modeled as three elements connected in parallel: a spring, a damper, and a contractile element, as shown in Fig. 1.

The functions of the three elements are given by

$$K^*(P) = K_0 + K_1 P \quad (1)$$

$$B(P) = B_0 + B_1 P \text{ (Inflation)} \quad (2)$$

$$B(P) = B_2 + B_3 P \text{ (Deflation)} \quad (3)$$

$$F(P) = F_0 + F_1 P \quad (4)$$

where K_0 , K_1 , B_0 , B_1 , B_2 , B_3 , F_0 , F_1 need to be identified by experiment; P is the internal pressure; $F(P)$ is the force exerted by the contractile element; $B(P)$ is the damping coefficient; and $K^*(P)$ is the spring coefficient.

The dynamic model for a novel PAMs-spring antagonistic structure has been established using the three-element model in [15], Fig. 1 illustrates the novel PAMs-spring antagonistic structure, in which the left side is a spring and the right side is a PAM. The total torque exerted by novel PAMs-spring antagonistic structure is shown in Eq. (5)

$$\tau = \left(F(P) - \frac{B(P)\dot{\theta}r\pi - K^*(P)\theta r\pi}{180} \right) r. \quad (5)$$

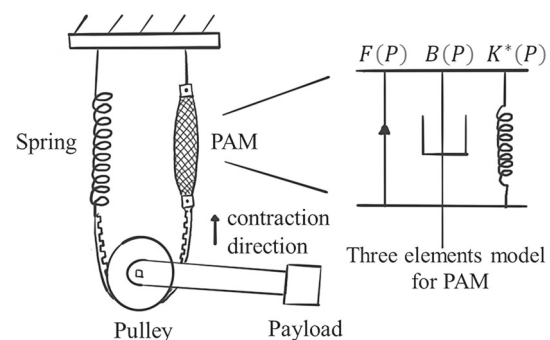


Fig. 1 Dynamic model for novel PAMs-spring antagonistic structure

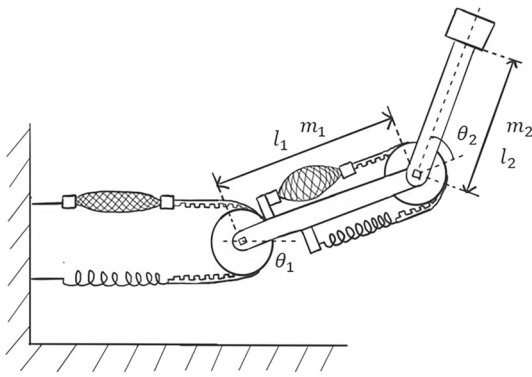


Fig. 2 PAMs-spring driven 2-DoF upper limb prostheses

Substituting Eqs. (1)–(4) into Eq. (5) and simplifying, the state-space function is obtained as follows:

$$\dot{\theta}_1 = \theta_2 \quad (6)$$

$$\dot{\theta}_2 = \frac{2[(F_0 - F_L) - (K_0 + K_L)\pi\theta_1 r/180 - B_0\theta_2 r\pi/180 + (F_1 - K_1\theta_1 r\pi/180 - B_1\theta_2 r\pi/180)u]}{mr} - \frac{2MgL\sin\theta_1}{mr^2} \quad (7)$$

where M is the mass of payload; L is the rod length; m is the pulley mass; r is the pulley radius; K_L is spring constant of left pre-tensioned springs, F_L is pre-tensioned force made by left side pre-tensioned spring; θ_1 is the rotation angle, θ_2 is the derivative of θ_1 ; u is output of controller, which means air pressure.

Based on Eqs. (6) and (7), a discretization of state-space function can be described as Eq. (8)

$$\theta(i+1) = (I + AT)\theta(i) + BTu(i) + TC(i) \quad (8)$$

where

$$A = \begin{pmatrix} 0 & 1 \\ \frac{-2(K_0+K_L)}{m} & \frac{-2B_0}{m} \end{pmatrix}, B = \begin{pmatrix} 0 \\ \frac{2(F_1-K_1\theta_1 r-B_1\theta_2 r)}{mr} \end{pmatrix},$$

$$C = \begin{pmatrix} 0 \\ \frac{2(F_0-F_L)}{mr} - \frac{2MgL\sin\theta_1}{mr^2} \end{pmatrix}, i \text{ is the number of iterations,}$$

I is identity matrix, T is time step.

Dynamic model of novel PAMs-spring driven 2-degrees of freedom (DoF) upper limb prostheses

To verify that the proposed method can be generalized to higher DoF, the dynamic model of a PAMs-spring driven 2-DoF upper limb prostheses is proposed, depicted in Fig. 2.

The dynamics of a 2-link rigid robot is given by Euler–Lagrange equation:

$$\tau = M_m(\theta)\ddot{\theta} + V(\theta, \dot{\theta}) + G(\theta) \quad (9)$$

where $M_m(\theta)$ is mass matrix, $V(\theta, \dot{\theta})$ is the Coriolis and centrifugal matrix, $G(\theta)$ is the gravity matrix. Substituting Eq. (5) into Eq. (9) and simplifying, the state-space function of the PAMs-spring driven 2-DoF upper limb prosthesis is obtained as follows:

$$\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = M_m(\theta)^{-1}(A + BP - V - G) \quad (10)$$

where

$$A = \begin{bmatrix} F_0 - \frac{\pi K_0\theta_1 r}{180} - \frac{\pi B_0 r\dot{\theta}_1}{180} - \frac{\pi K_L\theta_1 r}{180} \\ F_0 - \frac{\pi K_0\theta_2 r}{180} - \frac{\pi B_0 r\dot{\theta}_2}{180} - \frac{\pi K_L\theta_2 r}{180} \end{bmatrix},$$

$$B = \begin{bmatrix} F_1 - \frac{\pi K_1\theta_1 r}{180} - \frac{\pi B_1 r\dot{\theta}_1}{180} \\ F_1 - \frac{\pi K_1\theta_2 r}{180} - \frac{\pi B_1 r\dot{\theta}_2}{180} \end{bmatrix},$$

$$V = \begin{bmatrix} -m_1 l_1 l_2 \sin\theta_2 \dot{\theta}_1^2 - 2m_2 l_1 l_2 \sin\theta_2 \dot{\theta}_1 \dot{\theta}_2 \\ m_2 l_1 l_2 \sin\theta_2 \dot{\theta}_1^2 \end{bmatrix},$$

$$G = \begin{bmatrix} m_2 l_2 g \cos(\theta_1 + \theta_2) + (m_1 + m_2) l_1 g \cos\theta_1 \\ m_2 l_2 g \cos(\theta_1 + \theta_2) \end{bmatrix},$$

$$M_m = \begin{bmatrix} (m_1 + m_2) l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos\theta_2 & m_2 l_2^2 + m_2 l_1 l_2 \cos\theta_2 \\ m_2 l_2^2 + m_2 l_1 l_2 \cos\theta_2 & m_2 l_2^2 \end{bmatrix}.$$

l_1, l_2 are the lengths of the first and second links, m_1, m_2 are the masses of the first and second links, θ_1 and θ_2 are the joint angles of first and second links.

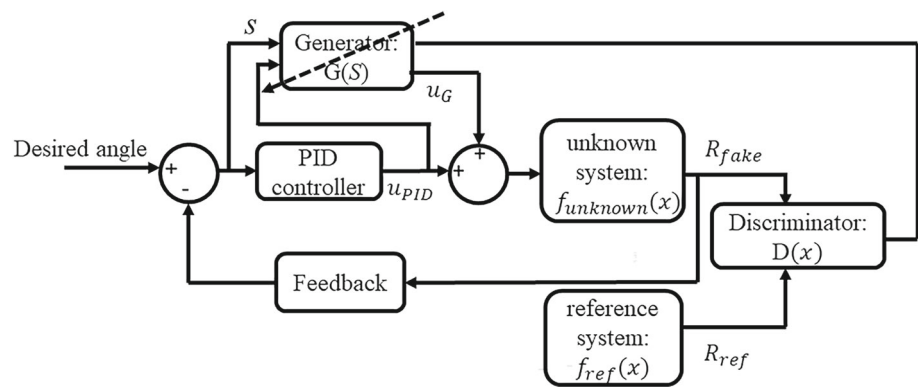
GAN and GAN-based PID

GAN is one of the generative models proposed in [33]. The generator G is a network that can capture the data distribution of training dataset and generate new data samples. The discriminator D can be used to estimate the probability that a sample comes from the training dataset rather than G . Through the GAN framework, G has the capability to generate new data samples that resemble the training dataset.

Moreover, the conventional PID controllers have been widely applied in industrial applications. The common equation is expressed as follows:

$$u_{PID} = K_P e(t) + K_I \int e(t) dt + K_D \frac{de(t)}{dt}$$

where u_{PID} is the output of PID controller, $e(t)$ is a feedback error, K_P, K_I, K_D is a proportional term, an integration term, and a derivative term, respectively.

Fig. 3 Control flowchart for GAN-based PID controller

To implement the GAN framework in control systems for adaptive control, we proposed a G-PID controller, the control flowchart is shown Fig. 3. Within the G-PID controller, the generator is input with system information S and produces a compensation action u_G , which is further input into the unknown but fixed system $f_{unknown}(x)$ combine with the PID control action u_{PID} . The unknown system produces a fake response R_{fake} . Concurrently, the discriminator D assesses the disparity between the fake response and reference response R_{ref} from reference system. Based on the comparison, the discriminator D updates the generator to optimize the compensation action. The G and D are updated through the loss function $V(G, D)$ shown in Eq. (11)

$$\begin{aligned} \min_{G, D} V(G, D) = & \mathbb{E}_{R_{ref} \sim p_{data}(R_{ref})} [\log D(R_{ref})] \\ & + \mathbb{E}_{fake \sim p_{data}(R_{fake})} \log[1 - D(f_{unknown}(G(s) + u_{PID}))]. \end{aligned} \quad (11)$$

The loss function $V(G, D)$ can be divided into four parts, which are: $\mathbb{E}_{R_{ref} \sim p_{data}(R_{ref})} [\log D(R_{ref})]$, $\mathbb{E}_{fake \sim p_{data}(fake)} \log[1 - D(f_{unknown}(G(s) + u_{PID}))]$, $\max_D V(G, D)$ and $\min_G V(G, D)$.

$$\mathbb{E}_{R_{ref} \sim p_{data}(R_{ref})} [\log D(R_{ref})] :$$

the reference response conforms the distribution $p_{data}(R_{ref})$. The term $\mathbb{E}_{R_{ref} \sim p_{data}(R_{ref})} [\log D(R_{ref})]$ represents the expected log-likelihood of R_{ref} as evaluated by discriminator D .

$$\mathbb{E}_{fake \sim p_{data}(R_{fake})} \log[1 - D(f_{unknown}(G(s) + u_{PID}))] :$$

by combining Eq. (11) with the Fig. 3, the generator G learns a mapping that transfers the collected system information S into the response space. In this process, the generator G initially transfers S to u_G , which can be expressed as

$$G : S \rightarrow u_G. \quad (12)$$

Subsequently, the u_G is input to an unknown but fixed mapping $f_{unknown}(x)$ and further transfer into the response space, which can be written as

$$f_{unknown} : u_G + u_{PID} \rightarrow R_{fake}. \quad (13)$$

Combining Eqs. (12) and (13), it is evident that in the process of the generator G transferring S into the response space, a u_{PID} term is included. This means that the solutions for generator G couples with u_{PID} , leading to a situation where multiple solution for u_G exist. However, in an extreme scenario, $u_{PID} = 0$, Eq. (13) can be expressed as

$$f_{unknown} : u_G \rightarrow R_{fake}. \quad (14)$$

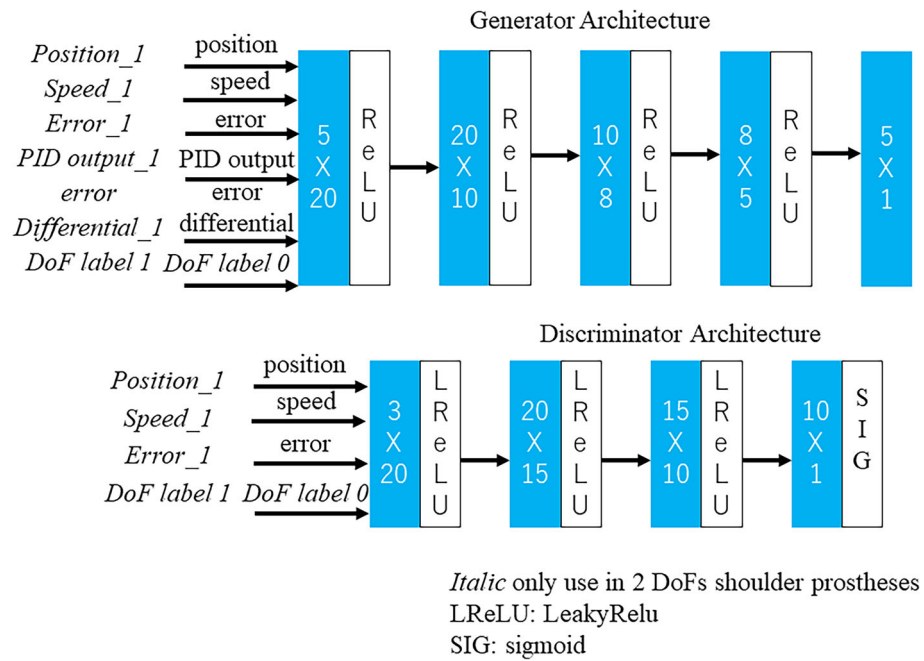
According to the composition of mappings, substituting Eq. (12) into Eq. (14) yields Eq. (15)

$$f_{unknown} \cdot G : S \rightarrow R_{fake}. \quad (15)$$

Equations (12)–(15) indicate the existence of at least one set of solutions, specifically when $u_{PID} = 0$, allowing u_G to directly control the unknown system to produce a response identical to that of the reference system. This implies that the generator has effectively learned how to compensate the unknown system's inverse dynamic model to that of the reference system. Therefore, generator G at least converges to this solution, rather than not converging at all. However, during the learning and optimization process, the GAN network may converge to other solutions as well. The Eqs. (12)–(15) are represented as $f_{unknown}(G(s) + u_{PID})$ in Eq. (11). $fake \sim p_{data}(fake)$ indicates that the fake response is sampled from $p_{data}(fake)$. The term $\mathbb{E}_{fake \sim p_{data}(R_{fake})} \log((1 - D(f_{unknown}(G(s) + u_{PID})))$ represents the expected log-likelihood of fake response as evaluated by discriminator D .

$$\max_D V(G, D) :$$

Fig. 4 The overall architecture of generator and discriminator



The objective of D is to completely differentiate between the fake response and reference response. In the ideal case for D, regardless of $f_{unknown}(G(s) + u_{PID})$ is, $D(f_{unknown}(G(s) + u_{PID}))$ should trend toward 0. Similarly, the output of $\log D(R_{ref})$ should approach to 0. In this scenario, the output of $V(G, D)$ should also converge to 0, representing its maximum output, indicates the best training of D.

$$\min_G V(G, D) :$$

the objective of G is to generate appropriate compensation action to control the unknown system, which in turn produces a fake response. The fake response intends to mislead D into making an incorrect judgment. In the case of an ideally functioning the G, the output of $D(f_{unknown}(G(s) + u_{PID}))$ should trend toward 1, and $\log((1 - D(f_{unknown}(G(s) + u_{PID})))$ should approach negative infinity. Thus, the minimization of $V(G, D)$ signifies the optimal training of the G.

The Eq. (11) represents a min–max game between the G and D. The D continuously strives to improve its ability to differentiate between fake and reference responses, while the G is engaged in enhancing compensation action that compensate the fake response identical to the reference response. Upon achieving convergence and reaching a Nash equilibrium, which indicates that both the G and D have enhanced their capabilities to a level where further improvements in either would not be possible without impacting the functional efficacy of the other [33]. At this point, the G is capable of compensating the unknown system to produces fake responses that are indistinguishable from

the reference responses, leads to the fake response distribution $p_{data}(R_{fake})$ becoming closer to the reference response distribution $p_{data}(R_{ref})$.

Figure 4 illustrates the architecture of the generator and discriminator in this paper. For a 1-DoF antagonistic joint, the inputs are position, speed, error, PID action, and error differential, which corresponds to the system information S in Fig. 3, and the output is u_g . When controlling a multi-DoF upper limb prosthesis, in addition to the previously mentioned 5 inputs, an additional input representing the DoF information is required.

For the discriminator, the inputs are position, speed, and error. The reason for selecting such three features as inputs is that the position, speed, and error of each step are unique to the reference system, these three indicators may be utilized to evaluate the consistency of the two system. Similarly, when controlling a system with multi-DoF, an additional input representing the DoF information is also necessary.

Both the generator and discriminator employ Multilayer Perceptrons (MLP). The generator consists of five layers, with ReLU activation functions used in the hidden layers. The discriminator has four layers, utilizing LeakyReLU (LReLU) between the input and hidden layers, and a sigmoid function in the output layer.

Simulation training process

The simulation training process could be summarized as follows: initially, the reference system must be chosen. In general, the response with the smallest overshoot with a faster response time could be chosen as the reference response.

Although not set in this paper, the choice of reference response also needs to consider specific constraints (such as maximum speed, overshoot, acceleration). Of course, theoretically, any response can be selected as the reference response. As shown in Fig. 3, at the t time, the current state (position, speed, and error) of the unknown system should be recorded, and the PID action along with the error differential need to be calculated based on the present state. Next, the generator is provided with unknown system's current state, the PID action u_{PID} , and the error differential. The generator generates the compensation action u_G . The combined actions of u_{PID} and u_G are fed into the unknown system, resulting a new fake state (which corresponds to time $t + 1$). Both the reference state and the new fake state are supplied to the discriminator. The discriminator and generator are then updated according to Eq. (11). Following an update, the new fake state is considered as the current state for the next iteration. The entire procedure is repeated until the control processes are complete.

During the training phase, the updating of the G and D using Eq. (11) necessitates routing the gradient through a function $f_{unknown}(x)$, which is the forward model of the unknown system. When a forward model is available (such as Eqs. (6) and (7)), it is possible to determine the gradient between G and D since the unknown system (including state space equations and payloads) is actually known throughout the training. This method is referred to model-based training process in this paper.

Alternatively, in the absence of a model, the gradient between G and D would vanish. To address this, a neural network, referred as the F-network in this paper, can replace the forward model. The input to the F-network comprises the combined actions of u_{PID} and u_G ; the output is the state at time $t + 1$. The F-network is continually trained at each timestep. Consequently, the F-network is comparable to simplified, discrete forward model of the unknown system. The gradient between G and D may be determined when the F-network converges; The F-neural network is a three-layer MLP with ReLU as the activation function. The input layer captures the output of G, and the output layer is designed to represent position, speed, and error. In this study, this strategy is referred to as the model-free training process.

Prototype online learning process

Even though the model has been trained in a simulation environment, as there are no constraints imposed in the simulation, there is still a discrepancy between the simulation and the prototype. Certain states achievable in the simulation may not be replicable in real-world, necessitating the implementation of prototype online learning based on the pre-trained model to overcome sim-to-real gap.

As mentioned above, in prototype experiments, the sensor in the prototype captures the states but do not provide gradients, necessitating the adoption of model-free training process for online learning. In the online learning, the F-network is employed, maintaining the same architecture as previously introduced. The labels are composed of the prototype's position, speed, and error recorded by sensor. The loss function is the MSE (Mean Squared Error) between the output of the F-neural network and the label. Prior to each update of the generator G, the F-neural network is modified 200 times. The loss function value below 1 post-update signifies convergence of the F-network, which then leads to subsequent updates of the G network. Conversely, the loss function value exceeding 1 indicates non-convergence, and the update for the G network is skipped. Since the aim of the F-neural network is to retrieve the gradient and it would be updated each time, overfitting is not a concern. The pseudocode is shown in Table 1.

System description

Simulation setup and training

In the previous research, the parameters of the PAMs' dynamic model [34] have been determined, as shown in Table 2. All the values of the PAMs' coefficients were obtained through measurements of the pneumatic actuator PM-10P (SQUSE), which weights 3 g for each, can provide a high output force (maximum tension of 100 N), and has an average shrinkage ratio of 30% per piece (around 2 cm for each). The negative sign of B_2 and B_3 indicates movement in the direction opposite to the positive. The pulley used in this study was Model P20MXL6.4B-HA-C pulley (Mitsubishi Belting Ltd., Japan), of which mass is about 10 g and radius r is 5 mm. With the results of an identification experiment, the time step T of the simulation was set to 0.03 s for 1-DoF joint case and 0.015 s for 2-DoF shoulder prosthesis. All the models were implemented using Python 3.9.12 64-bits. The neural network architectures were implemented by PyTorch '1.11.0 + cpu'. The optimizer of generator, discriminator and F-network was Adam optimizer with learning rate = $5e-4$, $1e-4$, and $1e-2$, respectively. The PID controller gains for 1-DoF joint was selected as $K_P = 0.25$, $K_I = 0.14$, and $K_D = 3.5$ and for 2-DoF shoulder prosthesis was selected as $K_P = 0.24$, $K_I = 0.001$, and $K_D = 0.001$ by try and error.

During the 1-DoF joint simulation training process, a system with 30 g payload was designated as the reference system, with its response serving as the reference response. Systems with payloads of 0 and 80 g were considered as unknown systems, with their responses treated as fake responses. One point worth noting is that, despite their designation as the unknown system, the payload M was known

Table 1 Pseudocode for the G-PID controller**Algorithm:** GAN based PID controller

Initialize G, D, F(*) network, select a reference system

Run the reference system and record all state information as reference response

1: **For** epochs $\rightarrow$ max epochs **do**

2: **For** T $\rightarrow$ max control time **do**

3: **For** iterations $\rightarrow$ max iterations **do (train D)**

4: Run unknown system with G and record new state as fake response

 Train the discriminator on the reference response

 Train the discriminator on the fake response

5: **End For**

6: **End For**

7: **For** T $\rightarrow$ max control time **do**

8: **For** iterations $\rightarrow$ max iterations **do (train G)**

9: **For** F_epochs $\rightarrow$ max F iterations **do (*)**

10: Update F-network(*)

11: **End For**

12: Run unknown system with G and record new state.

 Train the generator on the new state.

13: **End for**

14: **End for**

15: **END ALL**

(*) only do if use model-free training

Table 2 Coefficient sets used for the simulation

Parameters	Value	Unit	Parameters	Value	Unit
F_0	15	N	B_3	− 3	Ns/m
F_1	200	N	m	10	g
K_0	40	N/m	R	5	mm
K_1	2	N/m	K_L	200	N/m
B_0	20	Ns/m	F_L	5	N
B_1	9	Ns/m	L	10	cm
B_2	− 8	Ns/m	r	5	mm
2-DoF l_1	5	cm	2-DoF l_2	5	cm
2-DoF m_1	5	g	2-DoF m_2	5	g

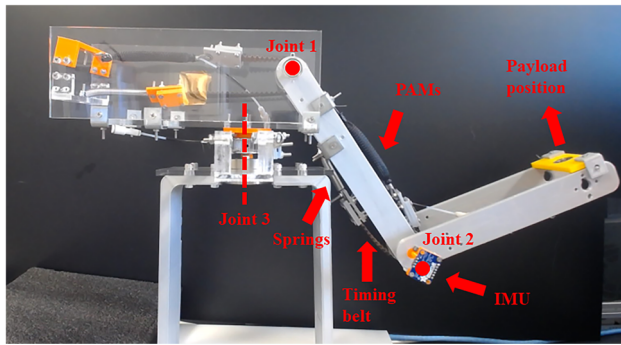


Fig. 5 Prototype of a 3-DoF upper limb prostheses

during the simulation training process but became unknown throughout the verification process. The three systems are controlled using the same PID controller gains, with the target angle set to 50° and the max control time set to 1.5 s (50 cycles). The maximum iterations for D training are 10. Following that, the G network is trained using the loss function shown in Eq. (11). The maximum iterations for G training are 5. The max epochs are 40.

Throughout the 2-DoF shoulder prosthesis simulation training process, a system with 35 g payload was selected as the reference system, while systems with payloads of 5 and 85 g were labeled as unknown systems. Training for the D is restricted to a maximum of 10 iterations, while the G is allowed up to 5 iterations. Maximum epochs allowed are 100.

Prototype setup and online learning

Figure 5 demonstrates the prototype of the upper limb prosthesis with 3-DoF. Despite the prosthesis contained 3-DoF, only joint 2 was utilized in this study. PAMs and a pre-tensioned spring constituted the antagonistic structure. During inflation, the PAMs drove the driving timing pulley, and its angle of rotation and angular velocity were conveyed to the controller through an inter-integrated circuit (IIC) bus by a BNO055 IMU sensor (Bosch Sensortec., Germany). Jetson nano (NVIDIA Corporation, USA) constituted the main control chip. The SMC ITV0030-2ML air proportional valve was utilized (SMC Pneumatics Ltd., Japan). Jetson nano transmitted angle and angular velocity to the computer and received a pulse-width-modulation (PWM) value over the serial connection in order to regulate the PAMs.

Prototype online learning was executed using the model learned from the simulation training experiment. During prototype online learning, the same PID gains were chosen as those used in the simulation experiment. The system with a 60 g payload in the prototype closely resembled the simulation's reference system, and therefore, was chosen as the reference system for online learning. The G-PID controller

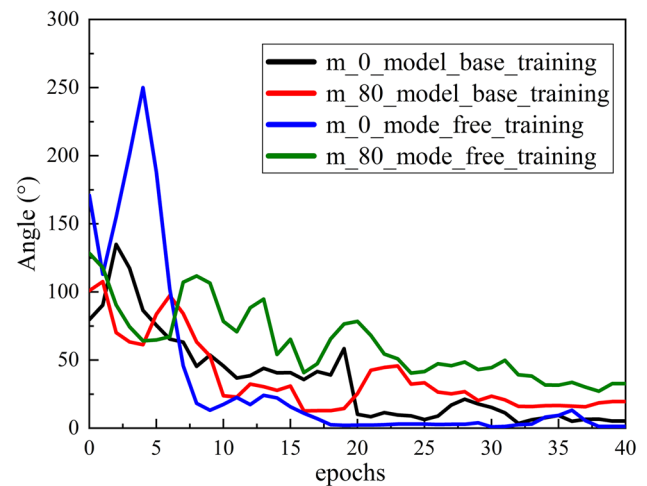


Fig. 6 Convergence graph

Table 3 Time consuming for different training method in real-world

	Model based training	Model free training
Time (s)	190	1327

was trained online for the cases of 0 and 80 g, and it was validated with 18 and 122 g.

Following is a description of the prototype online learning approach. The maximum iterations for D learning are 5. Following that, the F-network was updated by 200 times, next, the G network is trained using the Eq. 11. The maximum iterations for G learning is 3. The max epochs E is 1.

Results

Convergence process for 1-DoF joint

Two training methods, model-based and model-free, were utilized in the training 1-DoF joint process. Figure 6 depicts the MSE of the angle at each step between the reference system and the unknown system. It reveals whether it is the model-based or model-free training method, the system converges, although the ultimate error for the system with an 80 g payload is greater than that for the system with a 0 g payload. As demonstrated in the Table 3, although both methods execute 40 epochs, the time consumed varies.

The response under the G-PID controller in simulation for 1-DoF joint

After completing simulation training, the learned model is used to validate the systems with payloads of 0 and 80 g. At this stage, only the model trained using the model-based

Fig. 7 Simulation results 0 g vs 80 g (1-DoF joint)

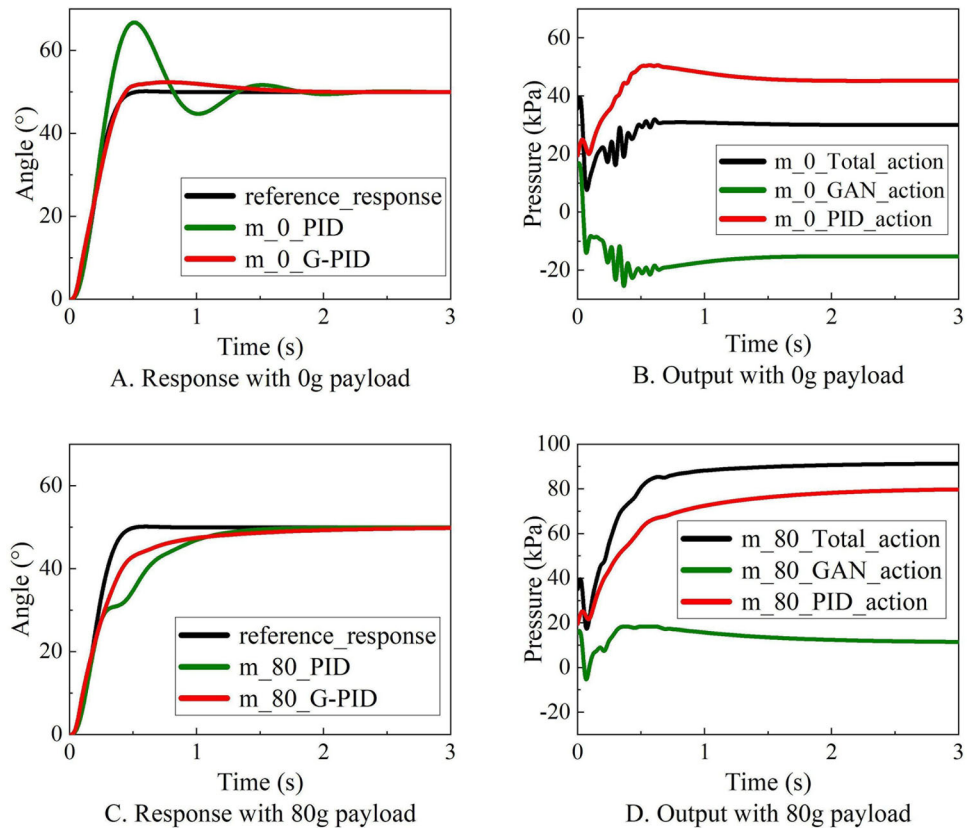


Table 4 System response under the G-PID (vs. PID) controller in simulation

Payload (g)	0	80	Reference
Overshoot (°)	2.35 (16.856)	0 (0)	0
Settling time (ms)	390 (780)	960 (1050)	390
Delay time (ms)	210 (190)	210 (260)	210
Rising time (ms)	270 (160)	730 (780)	300

training method are employed. Figure 7 and Table 4 illustrate the results.

In Fig. 7A, ‘reference response’ refers to user-preselected reference response, ‘m_0_PID’ means the response of the system with a 0 g payload using only-PID controller, ‘m_0_G-PID’ represents the response of the system with a 0 g payload using the G-PID controller. The definition of Fig. 7C is the same as (A). The results demonstrate that employing a trained GAN model as a compensator allows generator to create compensation action which can align the response of an unknown system with the reference response. When the payload of an unknown system is relatively light that led to overshoot, the generator generates a negative action to limit the system’s speed, hence preventing overshoot (Fig. 7B). Conversely, for heavier, slower systems, generator generates

a positive action that accelerates the system (Fig. 7D). The results in Table 4 show that after compensation by the generator, all the system performance metrics align more closely with the reference response.

To delve into the relationship between the generator’s input and output, the responses “m_0_G-PID” and “m_80_G-PID” were separated into three phases: from the start to the delay time (group 1), from the delay time to the rising time (group 2), and the period after the settling time (group 3). In addition, a comprehensive group 4 includes all data from the initial state to the end of the control process. It should be noted that Group 1 includes all delay time data from both the “m_0_G-PID” and “m_80_G-PID” responses, and similarly for the subsequent groups. Linear regression was employed to model the data within each group, described by Eq. (16), where c_1 , c_2 , c_3 , c_4 , and c_5 are coefficients pertaining to position, speed, error, PID action, and error differential. The bias is denoted by c_6 . Table 5 illustrates the regression results and the associated MSE upon convergence. In addition, the Spearman correlation coefficient and p-value were calculated between each generator input and output and are presented in Table 6.

$$\begin{aligned} \text{generator} = & c_1 \text{position} + c_2 \text{speed} + c_3 \text{error} \\ & + c_4 \text{pid}_{\text{action}} + c_5 \frac{\Delta e}{\Delta t} + c_6. \end{aligned} \quad (16)$$

Table 5 Fitting results

	Group 1	Group 2	Group 3	Group 4
c_1 (position)	0.1162	– 2.5226	– 1.0839	– 1.9939
c_2 (speed)	– 0.5825	– 0.6896	– 0.8816	– 0.53135
c_3 (error)	1.1691	1.2960	4.8136	– 0.03259
c_4 (PID action)	0.6497	2.1065	0.8089	1.6383
c_5 (error differential)	– 1.2831	– 3.7981	– 1.2112	– 4.2096
c_6 (bias)	0.7902	1.9438	1.2548	– 2.7509
MSE	0.065	6.0619	0.3543	37.9057

Bold highlights coefficient in each group

Table 6 Spearman rank correlation (p-value)

	Group 1	Group 2	Group 3
Position	0.12 (0.60)	– 0.03 (0.79)	– 0.91 (8.55e–21)
Speed	– 0.88 (2.58e–7)	– 0.55 (6.22e–5)	– 0.08 (0.56)
Error	0.12 (0.60)	0.03 (0.79)	0.91 (8.55e–21)
PID action	0.24 (0.29)	0.52 (2.31e–4)	0.84 (5.02e–5)
Error differential	0.50 (0.02)	0.51 (0.01)	0.08 (0.57)
Position + speed	– 0.80 (2.59e–21)	– 0.62 (1.11e–5)	– 0.45 (1e–3)

Bold highlights coefficient in each group

Table 5 shows that, with the exception of c_2 *speed, all coefficients in group1 are positive. It is important to notice that the error's differential is negative, hence c_5 * $\Delta e/\Delta t$ is also positive. In group 2, the coefficient for position also becomes negative, indicating that both position and speed influence the generator's output sign at this stage. Meanwhile, in group 3, other coefficients decrease or remain unchanged, the error coefficient c_3 becomes the largest, highlighting its significant contribution to this phase. Segmenting the responses into three groups results in considerably lower MSEs for each group compared to group 4, making it easy to assess the relationship between generator's input and output. Further tests were conducted with payloads of 10 and 60 g, as well as 20 and 50 g, yielding the same conclusions.

As shown in Table 6, the speed and output of the generator in group 1 have a strongest correlation, with an extremely low p-value underscoring the reliability of the data. Similarly, group 3 demonstrates the strongest correlation between error and generator output. Given that the error is calculated as “desired position – position” and the desired position remains constant, position's correlation coefficient is identical to that of error from the standpoint of the correlation coefficient.

Concerning the group 2 data, it can be noticed that the correlation between each feature and the output of the generator are relatively weak, with the strongest correlation is only 0.55 with speed. However, the linear regression analysis indicates that the output is reliant on both position and speed. Consequently, a series of further experiments were conducted to explore the relationship between “position + speed” and the

generator's output, revealing that the combined correlation of “position + speed” with generator output exceeds that of any single parameter, accompanied by a significantly low p-value.

In addition, the systems with payloads of 28 and 32 g were also controlled by the G-PID controller; the response and output are shown in Fig. 8. The response results indicate that when the difference between the unknown system and reference system is minimal, the G-PID controller does not play an effective compensatory effect. Specifically, for a 28 g payload, the generator should output a negative value, while for 32 g, it should return a positive value. However, for the 32 g case, the generator G outputs a negative value, indicating that the G-PID controller is incapable to make accurate compensation when the unknown system closely aligns with the reference system in Fig. 8D. Because the generator's output value is relatively small at this time, it exerts negligible influence on the system, with the PID controller primarily controlling the system. In addition, as represented in Fig. 7B, D and Fig. 8B, D, given that the initial state remains consistent, the input and output of the generator are also similar. This phenomenon is known as a cold start, which indicates the generator becomes active only when the system's is unique.

The response under the G-PID controller in simulation for 2-DoF shoulder prosthesis

Figures 9 and 10 depict the response of the 2-DoF prosthesis under payloads of 20, 60, 0, and 100 g. Here, 20 and 60 g are

Fig. 8 Simulation results 28 g vs 32 g (1-DoF joint)

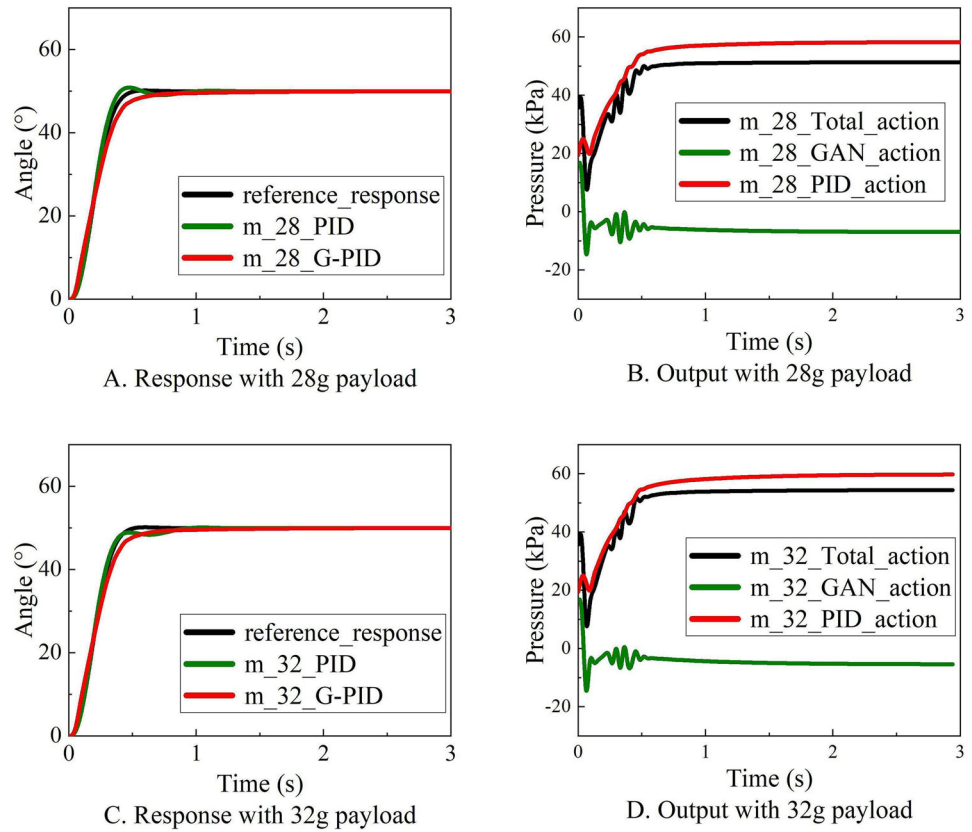


Fig. 9 Simulation results 20 g vs 60 g (2-DoF upper limb prosthesis)

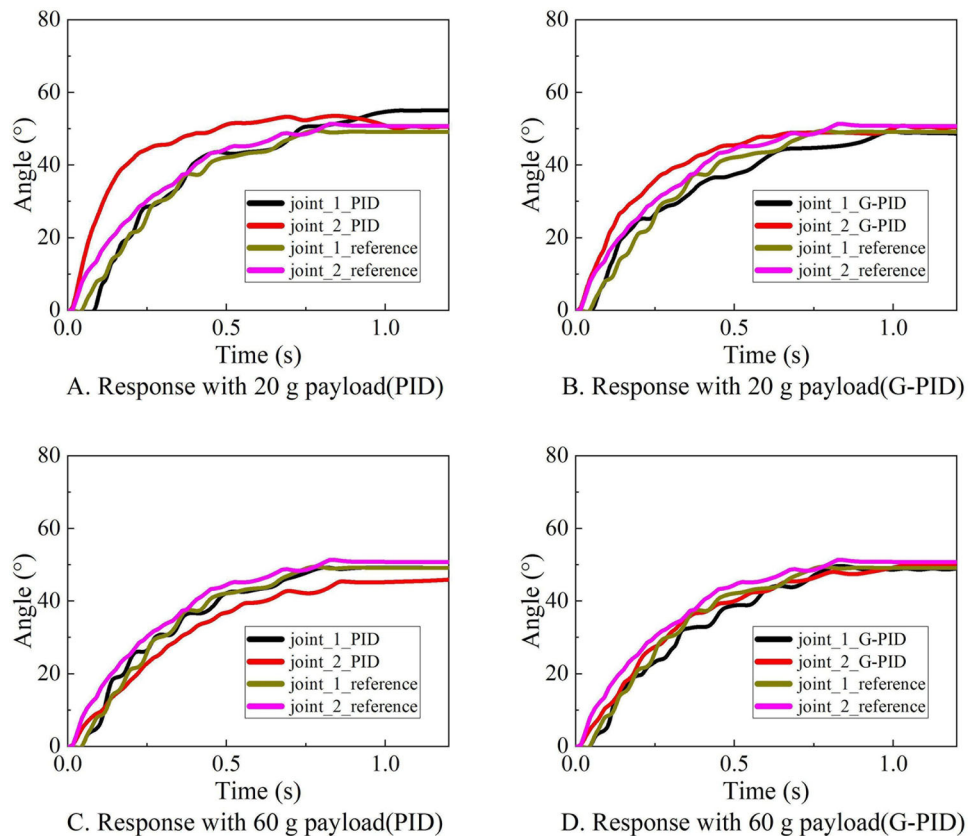
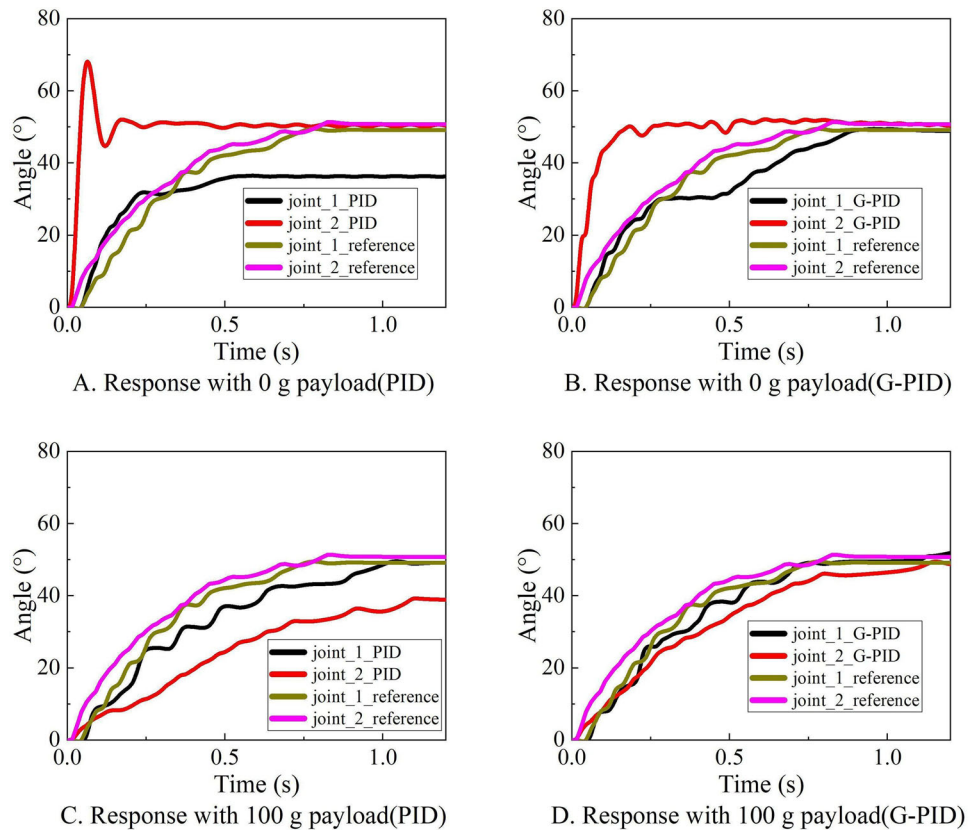


Fig. 10 Simulation results 0 g vs 100 g (2-DoF upper limb prosthesis)



payloads within the learning data range, whereas 0 and 100 g are payloads fall outside of the learning data range. All the results indicate that the G-PID controller has a compensatory effect on the control performance of the 2-DoF prosthesis.

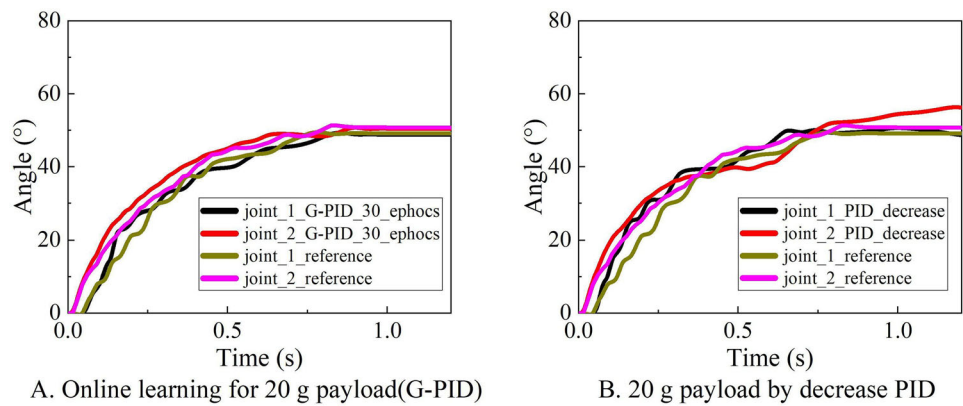
For a 20 g payload results in Fig. 9A, the initial stage of joint 2 is observed to be faster, whereas in the final stage of joint 1 has a steady-state error due to the relatively small integral term and the lengthy time required to erase the steady-state error. The G-PID controller is capable of decelerate the joint 2 and eliminate steady-state error in Fig. 9B. With a 60 g payload, the system's response is inherently close to the reference response, thus the G-PID controller provides a relatively low compensation value.

In the case of PID-only at 0 g payload in Fig. 10A, joint 2 has an extremely high overshoot, whereas joint 1 presents a steady-state error. The G-PID controller can also provide suitable compensation; however, although the overshoot of joint 2 is reduced, its response remains faster than the reference response. The overshoot of the joint 2's response is attributed to the fact that the initial output of the PID controller is predetermined based on its preset gains. This results in substantial state changes in a system without payload, causing an overshoot. Although the generator of the G-PID controller compensates the PID controller, its initial step output is fixed, rendering it incapable of compensating the system at this early stage. The steady-state error of joint 1 has

also been removed, while the rising process still exhibits a delayed response. With a 100 g payload, the G-PID controller enhances the system's responsiveness for both joints. Overall, these four instances encompass scenarios where both joints are slower than the reference response (60, 100 g), or one joint is faster while the other is slower compared to the reference response (0, 20 g) in PID-only case, with the G-PID controller accurately determining the direction of compensation value.

The 20 g group was used to further investigate G-PID's compensatory impact on the 2-DoF prosthesis. The 20 g system went through online training for 0.45 s with the result is shown in Fig. 11A. Compared to the results without online learning in Fig. 9B, the system with online learning aligns more closely with the reference response. Furthermore, regarding the decoupling ability of the G and D, Fig. 9A indicates that in the PID-only scenario, the response of joint 2 is faster than the reference response. To align the PID-only response closer to the reference response, a non-decoupling strategy was employed, which involved reducing the PID output for joint 2 by 3 kPa at each step. According to the results in Fig. 11B, although the speed of joint 2 decreases as predicted, the response of joint 1 likewise accelerates (at around 0.25 s). In addition, the overshoot occurs in the latter stage of joint 2 due to the increased error integral in the early stage as the speed diminishes. The G-PID controller reduces the

Fig. 11 Online learning and comparison for 20 g (2-DoF upper limb prosthesis)



speed of joint 2 without significantly affecting the behavior of joint 1, further illustrating the G-PID controller's capability for decoupling.

Prototype experiment results for 1-DoF joint

The prototype responses with the G-PID and PID controller are shown in Fig. 12 and Table 7 (refer to 'Supplementary Information 2, 3' for the video). The results demonstrate that when the varying payloads yield different response under the same PID controller gains, although the desired angle is ultimately tracked. This highlights the PID controller's lack of adaptability, as no single set of gains is effective across a wide range of payloads. With the G-PID controller, the overshoot of the system's response is eliminated, and the settling time is lowered from the maximum difference of 506.16 ms to 188.78 ms. Similarly, the delay time and rising time are reduced from maximum differences of 317.04 ms to 84.42 ms and 590.07 to 82.95 ms, respectively.

Additional experiment under the G-PID controller

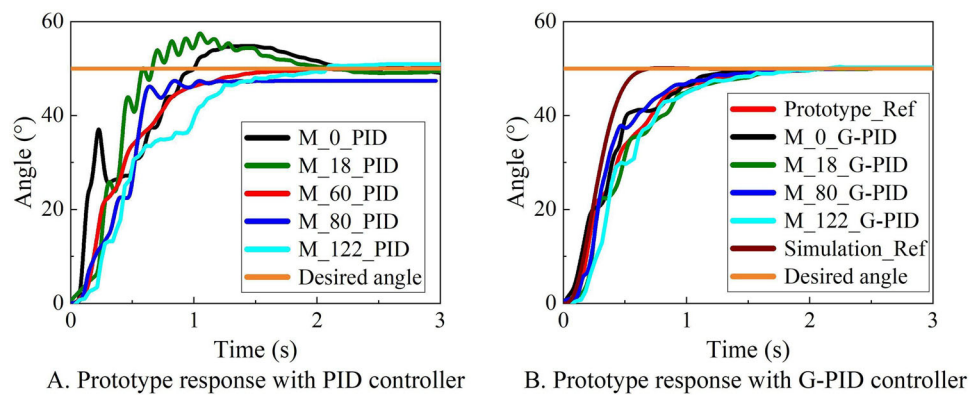
A comprehensive set of additional simulation and prototype experiments were undertaken to validate the G-PID controller's performance across a variety scenario including changes in initial conditions, target angle adjustments, external disturbances, and type of unknown parameters changing. For detailed experimental procedures, please refer to Supplementary Material 1.

Discussion

The adaptability of the controller is critical for control system, especially for an antagonistic structure driven by PAMs. The presence of unknown parameters with variable values within the control system significantly impacts the effectiveness of control.

To address such challenge, we present a GAN-based PID adaptive controller. The principle of the G-PID controller is that when a reference response is chosen, the generator can create a compensation action that aligns the unknown system's response closer to the reference response, thus enhancing the overall control adaptable. In this paper, to simplify the presentation of the experimental data and model, a conventional GAN network was utilized, both the generator and discriminator are simple MLPs.

Moreover, model-based and model-free training method were proposed, respectively. Figure 6 demonstrates that both methods ultimately converge with the convergence rate appearing almost identical. Since the model-free training incorporates the F-network, which requires pre-training prior to each training session for G, this increases the training duration to about 10 times that of the model-based training method. Furthermore, since the model-free training framework being $G \rightarrow F\text{-Network} \rightarrow D$, the system is more susceptible to overfitting and gradient vanishing. The convergence of the G-PID controller heavily relies on the convergence of the F-network. Conversely, the structure of the model-based training framework is $G \rightarrow \text{forward model} \rightarrow D$. This framework contains the physical information of the forward model, yielding more stable training and time efficiency. However, obtaining physical information is not always feasible, making the model-free training method more viable and suitable for non-experts and newcomers to the G-PID controller. Therefore, the choice between model-based and model-free training methods should be based on practical considerations, and we recommend model-based training when possible. The architecture of the F-Network should be based on practical considerations. In this paper, we only discuss systems with up to 2-DoF, for which a MLP can approximate the complexity of the robotic forward model. However, for unknown systems with higher DoF or in high-dimensional scenarios, considering a Convolutional Neural Network (CNN) or a more temporally aware architecture like Long Short-Term Memory (LSTM) networks could be beneficial.

Fig. 12 Prototype results**Table 7** System response under the G-PID (vs. PID) controller in prototype

Payload (g)	0	18	80	122
Overshoot (°)	0 (6.69)	0 (8.81)	0 (0)	0 (0)
Settling time (ms)	1044.96 (1863.41)	1233.74 (1633.34)	1116.76 (1357.25)	1220.92 (1464.85)
Delay time (ms)	387.80 (163.74)	430.69 (288.86)	346.27 (480.78)	390.12 (478.65)
Rising time (ms)	835.5 (772.28)	792.56 (350.62)	758.42 (517.61)	841.37 (940.69)

Figures 7 and 12 and Tables 4 and 7 illustrate the system's response under the control of the trained G-PID controller compared with the conventional PID controller in both simulation and prototype experiments. The results demonstrate that the G-PID controller's superior adaptability to the PID controller. Specifically, Fig. 7 illustrates that system with a 0 g payload and using only PID controller, the response exhibits significant overshoot and vibration. The implementation of the G-PID controller address these issues: the output of the generator turns negative, effectively reducing the overall output and stabilizing the system. For a system with 80 g payload, the response is rather sluggish and due to the system's rotational movement, the torque applied to the rotation center is angle dependent, lacking gravity compensation and resulting in a non-smooth transition process. Figures 7C and 12 illustrate how the application of the G-PID controller enhances the response characteristics, particularly by significantly smoothing out numerous non-smooth transition process. This enhancement is most notable when the system is subjected to a large payload and is transitioning between 60 to 90% of its steady state.

In simulation, the system's response properties, such as settling time and rising time does not change significantly, as shown in Table 4. However, prototype experiments show improvements in these aspects. It is possible that insufficient training contributes to the existence of such a circumstance. Moreover, as shown in Fig. 12A, due to a relatively high differential parameter in the PID controller, vibrations persist in prototype experiments. Conventionally, reducing such vibrations would require compromising control speed in PID

control theory, but the G-PID controller demonstrates an ability to reduce vibrations without affecting speed.

In addition, Fig. 7B, D indicate that the generator's output action is nonlinear. In general, determining how to compensate a system to modify its response to another specific response is challenging. While certain compensators, such as the lead compensator, can change the response of a system by ensuring that its root locus passes through a specified root, this approach is infeasible for complex systems. This implies that using compensation action as a label for training the compensator through supervised learning is not feasible. However, the G-PID controller can achieve the same function without the need for labels.

In Tables 5 and 6, the relationship between the input and output of the generator is analyzed to demonstrate how the G-PID controller can compensate the system only based on the system state, without considering the payload. Since all elements in delay time group (group 1) except c_2 *speed are positive, the generator's output symbol is determined by the speed. During the delay time phase, the system speed often varies significantly. With lighter payload, the system's speed increases, requiring the generator to generate a negative value to decelerate the system, and the opposite occurs for heavier loads. This behavior aligns with the real physical dynamics. In group 2, the position coefficient also turns negative. In cases where the system with lighter payload, the position throughout this process tends to be higher, prompting the generator to produce a negative value. In group 3, while other values either decrease or remain unchanged through the settling time, the error coefficient c_3 grows substantial,

indicating that the error plays a significant role in achieving steady state. A negative error indicates that the actual angle exceeds target, leads the generator output a negative value. Conversely, a position value indicates that the actual angle falls short of the target, prompts a positive output from the generator. These findings all corroborate that the G-PID controller is a state-based controller, and the output of the generator is related to the system state. In other words, the parameters that influence the training effectiveness of the G-PID controller must directly affect the system's response (no matter reference response or unknown response in training dataset). For example, in the reference system, the control gains of the PID controller directly impact the reference response; similarly, the varying payloads in the training dataset affect the fake response, which in turn influences the training results. On the other hand, the parameters that cannot directly influence the system's response, such as the types of unknown parameters in the training dataset (assuming two different parameters can yield the similar response), cannot affect the training effectiveness.

The simulation results for a 2-DoF shoulder prosthesis indicate the effectiveness of the G-PID controller to deal with multi-DoF robotic systems. Figures 9 and 10 demonstrate that the G-PID controller still determines the compensatory directions for various responses. However, the results in Fig. 9A indicate that the system responds well under PID-only control, even though joint 2's speed exceeds that of the reference response, it does not result in overshoot. Theoretically, this represents a superior response. However, upon implementing the G-PID controller, although the response becomes "worse", this phenomenon aligns with the G-PID controller concept. Therefore, the G-PID controller does not aim for the optimal response; rather it seeks an acceptable response that can be adjusted across a larger adaptation range. This highlights the significance of selecting the appropriate reference response, an aspect that warrants further investigation in future studies.

Figure 11A demonstrates that after compensation pattern are learned, further online learning can improve the accuracy of the compensation value. However, continuous online learning may lead to the problem of learning time and potential wear on the machinery. Moreover, based on the existing results, it is known that the G-PID controller can compensate the system to an acceptable level of performance even without online learning. Consequently, it's recommended to minimize the extent of online learning as much as possible. Additionally, a comparison between Fig. 11A and B reveals that the G-PID controller incorporates a certain decoupling ability. This means that G-PID's compensation for one joint takes into account the potential effects on the other, ensuring a balance across the system.

As illustrated in Figs. 7, 8, 9, 10, 11 and 12, all results are presented in terms of position. This approach is adopted

because the reference system used in this study is based on a position controller. However, since the aim of the G-PID controller is to align the response of the unknown system with that of the reference system, all metrics of the reference response (position, velocity, and acceleration) are desired targets. Therefore, even if the reference response becomes complex, the objective for the G-PID controller remains unchanged.

Consolidating all the results, two primary disadvantages of the G-PID controller have been pinpointed. The first issue is the "cold start", which is evident in Fig. 7B, D, where the G-PID controller cannot align the system's response to the reference response in the initial steps. If the two different system's starting state is same, then the compensation actions from G must also be uniform. Consequently, in the first step, the generator produces identical outputs for the systems with 0 g and 80 g payloads. However, following the first step, the generator begins to generate different compensation actions as the state of the system changes.

The second limitation is that the ineffectiveness of generator's compensation for systems closely resembling the reference system. This limitation is depicted in Fig. 8A, C, showing that when the unknown system closely resembles the reference system, the G-PID controller struggles to determine the appropriate compensation, potentially resulting in erroneous outputs. However, in such instances, the compensation value of generator tends to be minimal and only PID control may still obtain a satisfactory control performance. This presents a certain requirement for selecting the unknown systems for the training set: if the selected unknown systems in training dataset are inherently similar to the reference system, generator's training could end up being ineffective.

Conclusion and future work

In this paper, we introduce a GAN-based PID adaptive controller designed for a pneumatic artificial muscle-driven antagonistic structure. This controller stands out for its model-free and state-based nature, predictable control performance, and enhanced robustness. Through simulation and prototype experiments, we have not only validated its superior control adaptability on prostheses with either 1-DoF or 2-DoF but have also provided a physical analysis of generator's learning process. Furthermore, the additional experiments confirmed its generalizability, maintaining effectiveness across a range of scenarios, including variations in initial conditions, target angles, external disturbances, and the type of unknown parameters. These attributes make it particularly suitable for prosthetic control applications, hence facilitating their daily usage. Additionally, we are aware of the challenges faced by the G-PID controller, including the difficulty of distinguishing between two similar systems and the cold

start. These are problems that need to be addressed in future research.

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s40747-024-01488-y>.

Acknowledgements This work was partially supported by JST, the establishment of university fellowships towards the creation of science technology innovation, Grant Number JPMJFS2107 and Grant-in-Aid for Scientific Research (B), JSPS KAKENHI Grant Number 22H03450.

Data availability The original contributions presented in the study are included in the article and supporting material; further inquiries can be directed to the corresponding author.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Cordella F, Ciancio AL, Sacchetti R et al (2016) Literature review on needs of upper limb prosthesis users. *Front Neurosci* 10:1–14. <https://doi.org/10.3389/fnins.2016.00209>
- Furuya Y, Mikami T, Suzuki T, Fujimoto Y (2013) On an active prosthetic knee joint driven by a high thrust force helical motor. *IECON Proc (Ind Electron Conf)* 2:5894–5899. <https://doi.org/10.1109/IECON.2013.6700101>
- Niyetkaliyev AS, Hussain S, Ghayesh MH, Alici G (2017) Review on design and control aspects of robotic shoulder rehabilitation orthoses. *IEEE Trans Hum Mach Syst* 47:1134–1145. <https://doi.org/10.1109/THMS.2017.2700634>
- Mohseni O, Gagey F, Zhao G et al (2020) How far are Pneumatic Artificial Muscles from biological muscles? In: *Proceedings—IEEE international conference on robotics and automation*, pp 1909–1915. <https://doi.org/10.1109/ICRA40945.2020.9197177>
- Wang Y, Kokubu S, Zhou Z et al (2021) Designing soft pneumatic actuators for thumb movements. *IEEE Robot Autom Lett*. <https://doi.org/10.1109/LRA.2021.3105799>
- Zhang D, Wei B (2017) A review on model reference adaptive control of robotic manipulators. *Annu Rev Control* 43:188–198. <https://doi.org/10.1016/j.arcontrol.2017.02.002>
- Tsai TC, Chiang MH (2020) Design and control of a 1-DOF robotic lower-limb system driven by novel single pneumatic artificial muscle. *Appl Sci*. <https://doi.org/10.3390/app10010043>
- Zhou Z, Wang Y, Yu W (2022) The role of pre-tensioned springs in 3 pneumatic artificial muscles driven joint mechanisms with sliding mode controllers. *Robot Auton Syst* 151:104017. <https://doi.org/10.1016/j.robot.2021.104017>
- Marton L, Lantos B (2011) Control of robotic systems with unknown friction and payload. *IEEE Trans Control Syst Technol* 19:1534–1539. <https://doi.org/10.1109/TCST.2010.2086458>
- Dai S, Lee T, Bernstein DS (2014) Adaptive control of a quadrotor UAV transporting a cable-suspended load with unknown mass. In: *Proceedings of the IEEE conference on decision and control*. IEEE, Los Angeles, CA, pp 6149–6154
- Zhou Z, Wang Y, Yu W (2022) A simulation study for evaluating the role of pre-tensioned springs in 3 pneumatic artificial muscle driven joint mechanisms with sliding mode controllers. In: Ang MH, Asama H, Lin W, Foong S (eds) *Lecture notes in networks and systems*, vol 412. Springer, Cham, pp 343–354
- Ahmed AS, Kadhim SK (2022) A comparative study between convolution and optimal backstepping controller for single arm pneumatic artificial muscles. *J Robot Control* 3:769–778. <https://doi.org/10.18196/jrc.v3i6.16064>
- Bou Saba D, Massioni P, Bideaux E, Brun X (2019) Flatness-based control of a two degrees-of-freedom platform with pneumatic artificial muscles. *J Dyn Syst Meas Control Trans ASME* 141:1–8. <https://doi.org/10.1115/1.4041445>
- Šitum Ž, Herceg S, Bolf N, Ujević Andrijić Ž (2023) Design, construction and control of a manipulator driven by pneumatic artificial muscles. *Sensors* 23:1–14. <https://doi.org/10.3390/s23020776>
- Zhou Z, Kokubu S, Wang Y et al (2022) Optimization of spring constant of a pneumatic artificial muscle-spring driven antagonistic structure. *IEEE Robot Autom Lett* 7:5982–5989. <https://doi.org/10.1109/lra.2022.3162021>
- Zhu X, Tao G, Yao B, Cao J (2009) Integrated direct/indirect adaptive robust posture trajectory tracking control of a parallel manipulator driven by pneumatic muscles. *IEEE Trans Control Syst Technol* 17:576–588. <https://doi.org/10.1109/TCST.2008.2001715>
- Sun N, Di Liang Wu Y et al (2020) Adaptive control for pneumatic artificial muscle systems with parametric uncertainties and unidirectional input constraints. *IEEE Trans Ind Inform* 16:969–979. <https://doi.org/10.1109/TII.2019.2923715>
- Tahri A, El Fadil H, Giri F, Chaoui FZ (2016) Nonlinear adaptive control of a hybrid fuel cell power system for electric vehicles—a Lyapunov stability based approach. *Asian J Control* 18:166–177. <https://doi.org/10.1002/asjc.1231>
- Peternel L, Noda T, Petrič T et al (2016) Adaptive control of exoskeleton robots for periodic assistive behaviours based on EMG feedback minimisation. *PLoS ONE* 11:1–26. <https://doi.org/10.1371/journal.pone.0148942>
- Kiguchi K, Kariya S, Watanabe K et al (2001) An exoskeletal robot for human elbow motion support—sensor fusion, adaptation, and control. *IEEE Trans Syst Man Cybern Part B Cybern* 31:353–361. <https://doi.org/10.1109/3477.931520>
- Li M, Deng J, Zha F et al (2018) Towards online estimation of human joint muscular torque with a lower limb exoskeleton robot. *Appl Sci*. <https://doi.org/10.3390/app8091610>
- Bansal HO, Sharma R, Shreeraman PR (2012) PID controller tuning techniques: a review. *J Control Eng Technol* 2:168–176
- Anh HPH (2010) Online tuning gain scheduling MIMO neural PID control of the 2-axes pneumatic artificial muscle (PAM) robot arm. *Expert Syst Appl* 37:6547–6560. <https://doi.org/10.1016/j.eswa.2010.02.131>
- Nuchkrua T, Leephakpreeda T (2013) Fuzzy self-tuning PID control of hydrogen-driven pneumatic artificial muscle actuator. *J Bionic Eng* 10:329–340. [https://doi.org/10.1016/S1672-6529\(13\)60228-0](https://doi.org/10.1016/S1672-6529(13)60228-0)
- Liu H, Li Y, Zhang Y et al (2018) Intelligent tuning method of PID parameters based on iterative learning control for atomic force

- microscopy. *Micron* 104:26–36. <https://doi.org/10.1016/j.micron.2017.09.009>
26. Cao Y, Huang J (2020) Neural-network-based nonlinear model predictive tracking control of a pneumatic muscle actuator-driven exoskeleton. *IEEE/CAA J Autom Sin* 7:1478–1488. <https://doi.org/10.1109/JAS.2020.1003351>
 27. Ulkir O, Akgun G, Kaplanoglu E (2019) Real-time implementation of data-driven predictive controller for an artificial muscle. *Stud Inform Control* 28:189–200. <https://doi.org/10.24846/v28i2y201907>
 28. Hou Z, Xiong S (2019) On model-free adaptive control and its stability analysis. *IEEE Trans Autom Control* 64:4555–4569. <https://doi.org/10.1109/TAC.2019.2894586>
 29. Franco E, Casanovas AG, Rodriguez Baena YF, Astolfi A (2019) Model based adaptive control for a soft robotic manipulator. In: *Proceedings of IEEE conference on decision and control* 2019-December, pp 1019–1024. <https://doi.org/10.1109/CDC40024.2019.9029449>
 30. Tang ZQ, Heung HL, Tong KY, Li Z (2021) Model-based online learning and adaptive control for a “human-wearable soft robot” integrated system. *Int J Robot Res* 40:256–276. <https://doi.org/10.1177/0278364919873379>
 31. Miyamoto H, Kawato M, Setoyama T, Suzuki R (1988) Feedback-error-learning neural network for trajectory control of a robotic manipulator. *Neural Netw* 1:251–265. [https://doi.org/10.1016/0893-6080\(88\)90030-5](https://doi.org/10.1016/0893-6080(88)90030-5)
 32. Napole C, Barambones O, Calvo I, Velasco J (2020) Feedforward compensation analysis of piezoelectric actuators using artificial neural networks with conventional PID controller and single-neuron PID based on Hebb learning rules. *Energies* 13:1–16. <https://doi.org/10.3390/en13153929>
 33. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. *Adv in Neural Inf Process Syst*, Montreal, Canada, December 08–13, 2014. MIT Press, Cambridge, pp 2672–2680
 34. Reynolds DB, Repperger DW, Phillips CA, Bandry G (2003) Modeling the dynamic characteristics of pneumatic muscle. *Ann Biomed Eng* 31:310–317. <https://doi.org/10.1114/1.1554921>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.